Chapter 8
Prototype System Based on Heterogeneous Network

Abstract Because of significant advantages of heterogeneous information network, it is widely used to model networked data, and many data mining tasks have been exploited on it. Besides that, many prototype systems, even real systems, have been built based on heterogeneous networks. In these systems, heterogeneous networks are constructed, stored, and operated based on real networked data, and many novel applications are designed based on heterogeneous networks. In this chapter, we introduce two prototype systems for recommendation and further give a brief review on other systems based on heterogeneous networks.

8.1 Semantic Recommender System

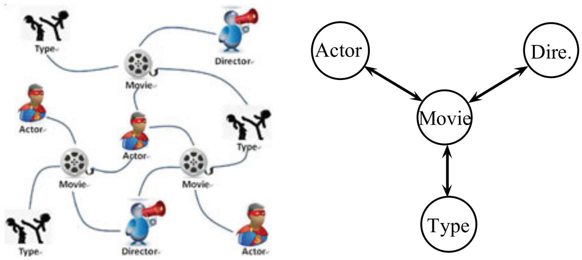
8.1.1 Overview

Many recommendation methods have been proposed, which can be roughly classified into two categories: content-based filtering (CB) and collaborative filtering (CF). CB analyzes correlations between the content of the items and the user's preferences [1]. CF analyzes the similarity between users or items [2]. These methods have been applied to recommender systems and achieved great success. However, these recommender systems may have the following disadvantages.

- Conventional recommender systems usually recommend similar products to users without exploring the semantics of different similarity measures. However, the similar products are often different based on similarity semantics. For example, in the movie recommendation, the similar movies based on the same actors are different from those based on the same directors. Conventional systems usually give a recommendation without considering the subtle implications of similarity semantics. The proposed system is more appealing to provide a semantic recommendation function, which will give more accurate recommendation when users know their intents.
- Conventional systems only recommend same-typed objects. However, a system may be more useful if it simultaneously recommends more related objects under different semantics. For example, when users select movies, the system not only

C. Shi and P.S. Yu, *Heterogeneous Information Network Analysis and Applications*, Data Analytics, DOI 10.1007/978-3-319-56212-4_8

Fig. 8.1 An example of heterogeneous information network and its schema



(a) Heterogeneous network of movie data (b) Network schema

recommends the similar movies, but also suggests some related actors and directors (note that they are not limited to the actors and directors of this movie). The user may find an interesting actor and then search the movies of the actor. The relevance recommendation will provide richer information and enhance user experience.

Nowadays, social networks consisting of different types of information become popular. Particularly, the advent of the Heterogeneous Information Network (HIN) [4] provides a new perspective to design the recommended system. HINs are the logical networks involving multiple-typed objects and multiple-typed links denoting different relations. It is clear that HINs are ubiquitous and form a critical component of modern information infrastructure [4]. Although the bipartite network [8] has been applied to organize components of recommended system, HIN is a more general model which contains more comprehensive relations among objects and much richer semantic information. Figure 8.1a shows an HIN example on the movie recommendation data. The network includes the richer objects (e.g., movie, actor, director) and their relations. The network structure can be represented with the star schema as shown in Fig. 8.1b. HIN has an unique property [10, 14]: the different paths connecting two objects have different meanings. For example, in Fig. 8.1b, movies can be connected via "Movie–Actor–Movie" (*MAM*) path, "Movie–Type–Movie" (*MTM*) path, and so on. It is clear that the semantics underneath these paths is different. The *MAM* path means that movies have the same actors, while the *MTM* path means that movies of the same type. Here, the meta path connecting two-typed objects is defined as relevance path [10]. Obviously, the distinct semantics under different relevance paths will lead to different relatedness and recommendation.

Focusing on non-personalized recommendation, this chapter demonstrates a semantic recommended system, called HeteRecom. Different from conventional recommended systems, it is based on HIN. Generally, HeteRecom has the following unique features. (1) Semantic recommendation: The system can recommend objects of the designated type based on the relevance path specified by users. (2) Relevance recommendation: Besides the same-typed objects recommendation, the system can recommend other related objects.

The implementation of HeteRecom faces the following challenges. (1) Relevance measure of heterogeneous objects: In order to recommend the different type objects,

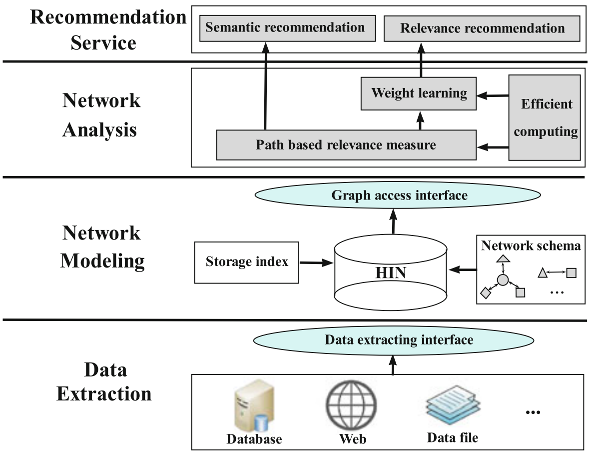


Fig. 8.2 The architecture of *HeteRecom* system

the system needs to measure the relatedness of different type objects. (2) The weight learning method: It is a key issue for an integrated recommendation to automatically determine the weights of different relevance paths. (3) Efficient computing strategies: In order to provide online service, the recommended system needs to efficiently compute the relevance measure. In order to solve these challenges, the HeteRecom system first applies a path-based relevance measure, which can not only effectively measure the relatedness of any-typed objects but also subtly capture the semantics containing in the relevance path. Besides, a heuristic weight learning method can automatically determine the weights of different paths. Moreover, many computing strategies are designed to handle huge graph data. This paper demonstrates the effectiveness of HeteRecom on the real movie data through providing online semantic and relevance recommendation services.

8.1.2 System Architecture

Figure 8.2 shows the architecture of HeteRecom, which mainly consists of four components:

- Data extraction: It extracts data from different data source (e.g., database and Web) to construct the network.
- Network modeling: It constructs the HIN with a given network schema. According to the structure of data, users can specify the network schema (e.g., bipartite, star, or arbitrary schema) to construct the HIN database. The database provides the store and index functions of the node table and edge table of the HIN.
- Network analysis: It analyzes the HIN and provides the recommendation services. It first computes and stores the relevance matrix of object pairs by the path-based relevance measure. Based on the relevance matrix and efficient computing

strategies, the system can provide the online semantic recommendation service. Through the weight learning method, it can combine the relevance information from different semantic paths and provide online relevance recommendation service.

- Recommendation service: It provides the succinct and friendly interface of recommendation services.

8.1.3 System Implementation

It is challenging in many ways to implement these components. First, it is difficult to measure the relatedness of any-typed objects in a HIN. Second, It is not easy to combine those recommendation information on different semantic paths. Third, there are many challenges in the computation and storage of huge relevance matrix. In the following section, we will present the solutions to these challenges.

8.1.3.1 A Path-Based Relevance Measure

We apply the HeteSim [10], a path-based relevance measure, to do semantic recommendation. The basic idea behind HeteSim is that similar objects are related to similar objects. The HeteSim is defined as follows:

Definition 8.1 (*HeteSim* [10]) Given a relevance path $P = R_1 \circ R_2 \circ \cdots \circ R_l$, HeteSim between two objects s and t ($s \in R_1.S$ and $t \in R_l.T$) is:

$$HeteSim(s, t|R_1 \circ R_2 \circ \cdots \circ R_l) = \frac{1}{|O(s|R_1)||I(t|R_l)|} \sum_{i=1}^{|O(s|R_1)|} \sum_{j=1}^{|I(t|R_l)|} HeteSim(O_i(s|R_1), I_j(t|R_l)|R_2 \circ \cdots \circ R_{l-1}) \tag{8.1}$$

where $O(s|R_1)$ is the out-neighbors of s based on relation R_1, $I(t|R_l)$ is the in-neighbors of t based on relation R_l, and $R.S$ ($R.T$) represents the source (target) object of relation R, respectively.

Essentially, $HeteSim(s, t|P)$ is a pairwise random walk-based measure, which evaluates how likely s and t will meet at the same node when s follows along the path and t goes against the path. The path implies the semantic information and *HeteSim* evaluates the relatedness of any-typed object pairs according to the given path. The *HeteSim* measure has shown its potential in object profiling, experts finding, and relevance search. The detailed information can be seen in [10].

Since relevance paths embody different semantics, users can specify the path according to their intents. The semantic recommendation calculates the relevance matrix with *HeteSim* and recommends the top k objects.

8.1.3.2 Weight Learning Method

There are many relevance paths connecting the query object and related objects, so the relevance recommendation should comprehensively consider the relevance measures based on all relevance paths. It can be depicted as follows:

$$Sim(A, B) = \sum_{i=1}^{N} w_i * HeteSim(A, B|P_i) \tag{8.2}$$

where N is the number of relevance paths, P_i is a relevance path connecting the object types A and B, w_i is the weight of path P_i. Although there can be infinite relevance paths connecting two objects, we only need to consider those short paths, since the long paths are usually less important [14].

The next question is how to determine the weight w_i. The supervised learning [7] can be used to estimate these parameters. However, it is impractical for an online system: (1) It is time-consuming, even impractical, to learn these parameters on an online system. (2) It is a very labor-intensive and subjective work to label those learning instances. Here, we propose a heuristic weight learning method.

The importance (I) of a path $P = R_1 \circ R_2 \circ \cdots \circ R_l$ is determined by its strength (S) and length (l). Obviously, the path strength is decided by the strength of relations constructing the path, which can be defined as follows:

$$S(P) = \prod_{i=1}^{l} S(R_i) \tag{8.3}$$

The strength of a relation $A \xrightarrow{R} B$ is related to the degree of A and B based on R. Intuitively, if the mutual connective links between A and B are smaller, they are more important to each other, so their relation strength is stronger. For example, the relation strength between movie and director (MD) is stronger than that between movie and type (MT). So we can define the relation strength as follows:

$$S(R) = (O(A|R)I(B|R))^{-\alpha}(\alpha \in [0, 1]) \tag{8.4}$$

where $O(A|R)$ is the average out-degree of type A and $I(B|R)$ is the average in-degree of type B based on relation R.

The importance (I) of the path P is positively correlative to the path strength (S) and negatively correlative to the path length (l). Here, we define it as follows:

$$I(P) = f(S, l) = e^{S-l} \tag{8.5}$$

For multiple paths $(P_1, P_2, \cdots, P_N)$, the weight w_i of path P_i is

$$w_i = \frac{I_i}{\sum_{i=1}^{N} I_i} \tag{8.6}$$

In HeteRecom, we consider all relevance paths whose length is smaller than a threshold *Len*. The relevance recommendation combines the relevance measure results of all these paths with the weight learning method and makes an integrated recommendation.

8.1.3.3 Efficient Computing Strategies

As an online recommended system, *HeteRecom* needs to do a real-time recommendation for user's query. However, an HIN is usually huge and the computation of HeteSim is time-consuming. So the system employed many efficient computing strategies. Three basic strategies are depicted as follows:

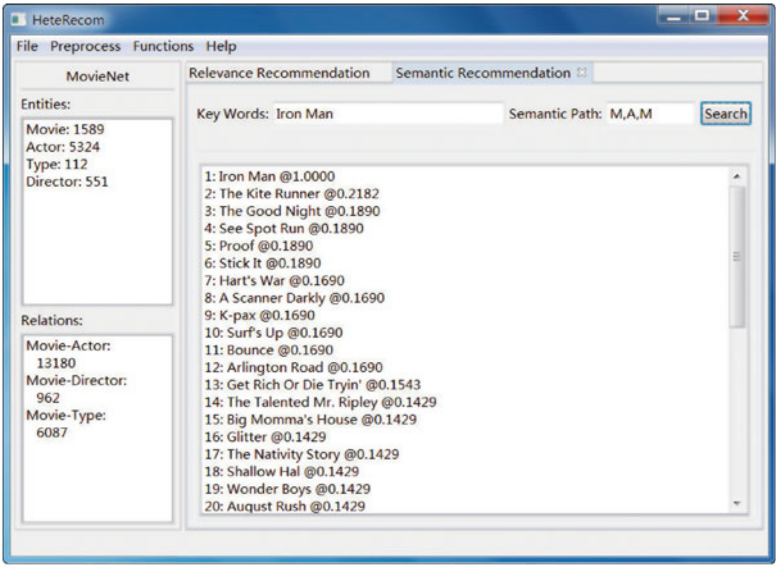
Off-line computation: The primary strategy is to compute relevance matrix offline and make recommendations online. For frequently used relevance paths, the relevance matrix $HeteSim(A, B|P)$ can be calculated ahead of time. The online recommendation on $HeteSim(a, B|P)$ will be very fast, since it only needs to locate the position in the matrix.

Fast matrix multiplications: The most time-consuming component in the system is the matrix multiplications in HeteSim. There are many frequent patterns in relevance paths. Since the matrix multiplications satisfy the associative law, we can precede to compute the product of frequent patterns iteratively. Moreover, those frequent patterns only need to be computed once. For example, we only need to compute the frequent pattern *AMA* once for the symmetric path *AMAMA*. Since the short pattern is more frequent, we only find the most frequent relation pair in each iteration.

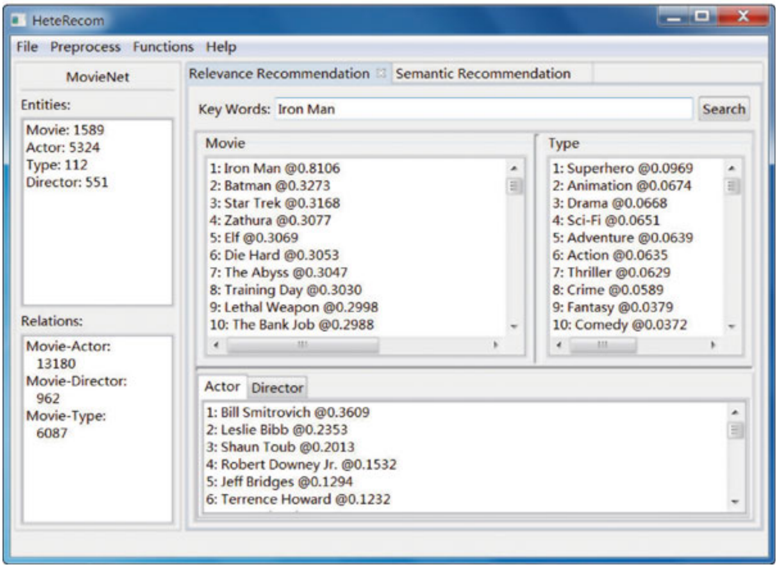
Matrix sparsification: The relevance matrix often becomes denser along the matrix multiplications [7]. The dense matrix may cause two difficulties. (1) Matrix multiplications cost a lot of time and space. (2) It costs a lot of time and huge memory to load and search these dense relevance matrix. As a consequence, we need to sparsify the reachable probability matrix along the matrix multiplications without much loss of accuracy. The basic idea is to truncate those less important nodes whose relevance value is smaller than a threshold ε. The static threshold [7] is not suitable, since it may truncate some important nodes with small relevance values and keep those unimportant nodes with large relevance values. Since we usually pay close attention to the top k recommendation, we set the threshold ε as the top k relevance value of the matrix. The k is dynamically adjusted as follows:

$$k = \begin{cases} L & \text{if } L \leq W \\ \lfloor (L-W)^{\beta} \rfloor + W (\beta \in [0, 1]) & \text{others} \end{cases}$$

where L is the vector length. W is the threshold which determines the size of nonzero elements. The larger W or β may lead to the denser matrix with less loss. In order to



(a) Semantic recommendation based on *MAM* path



(b) Relevance recommendation

Fig. 8.3 The HeteRecom prototype system quickly determine the top k relevance value, it is approximately computed with the sample data from the raw matrix.

8.1.4 System Demonstration

We showcase the HeteRecom prototype system using IMDB movie data as the example application. The IMDB movie data was downloaded from The Internet Movie Database.[1] The IMDB movie data collects 1591 movies before 2010. The related objects include actors, directors, and types, which are organized as a star schema shown in Fig. 8.1b.

Figure 8.3 demonstrates the interface of the HeteRecom system, which is developed with Java. The left part of interface shows the basic information of the dataset. The right part shows the recommendation results. In the semantic recommendation, users specify the key words and semantic path, the recommendation results will be exhibited in the panel. Figure 8.3a shows the movies with the same actors of "Iron Man" by specified the "MAM" path. The *HeteRecom* can make many recommendations that conventional systems cannot do. For example, recommending the movies that have the same style with the movies of "Arnold Schwarzenegger" can be done by the path $AMTM$. In the relevance recommendation, the system can simultaneously recommend different-typed objects. Figure 8.3b shows the recommendation results of the movie "Iron Man," which include the similar movies and related actors, directors, and types. We can make many interesting recommendations on *HeteRecom*. For example, if we want to know the information about the action movie, we can search "action." The system will recommend related action movies, actors, and directors.

8.2 Explainable Recommender System

8.2.1 Overview

In order to tackle the information overload problem on WWW, many recommendation techniques have been proposed to build recommender systems. These recommended systems have been widely applied to e-commerce companies and achieved great success, for example, the book recommendation in Amazon and movie recommendation in Netflix. However, the explanation of recommendation results is a very important but seldom exploited problem. Good explanations could help inspire user trust and loyalty and increase satisfaction. Recommendation explanation makes it quicker and easier for users to find what they want and persuade them to try or purchase a recommended item [20]. Contemporary explanations of recommendations

[1] www.imdb.com/.

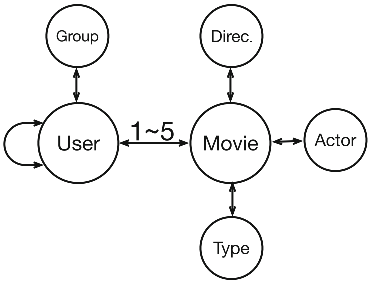


Fig. 8.4 Network schema of HIN constituted by Douban movie recommendation

usually use features or characteristics of users or the recommended item as intermediary entities. For example, the MoviExplain system employs movie features to justify recommendations [15], and Vig et al. [21] design Tagsplanations to provide explanation based on community tags. With the surge of social recommendation, there are some works on social explanation. Wang et al. [22] propose an algorithm to generate the most persuasive social explanation; Sharma et al. [9] present a study of the effects of social explanations in a music recommendation context. These methods try to explain recommendation through one type of information (e.g., features or social relations), while the recommendation results may stem from complex heterogeneous information and various factors. The recommended system needs to explain these factors more clearly.

In this chapter, we develop a **Rec**ommender system with **Exp**lanation (called RecExp). Inspired by the recent surge of heterogeneous information network [11], we organize the objects and relations in a recommended system as an HIN. Figure 8.4 shows such an example in movie recommendation. The HIN not only contains different types of objects in movie recommendation (e.g., users and movies) but also illustrates all kinds of relations among objects, such as viewing information, social relations, and attribute information. Moreover, two objects in an HIN can be connected via different paths, called meta path, and different meta paths have different meanings. So we can find the similar users of a user through different meta paths connecting these two users, and then we can combine the recommendation results of different similar users under different meta paths. Based on this idea, we design the semantic recommended system, RecExp, with explanation, which has the following two significant features:

- Semantic recommendation: Utilizing different meta paths, RecExp can find different similar users, and thus generate different recommendation results according to these similar users. Moreover, these meta paths correspond to different recommendation models, so RecExp can realize semantic recommendation through selecting proper meta paths.
- Recommendation explanation: RecExp utilizes semantics and weights of meta paths to present personalized recommendation explanation, which can reveal user preferences and make explanation more persuasive.

8.2.2 Heterogeneous Network-Based Recommendation

In this section, we will briefly introduce the basic concept and method used in RecExp. HIN [11] is a special type of information network with the underneath data structure as a directed graph, which contains either multiple types of objects or multiple types of links. Objects and their relations in recommended system constitute an HIN. Figure 8.4 shows the network schema of the movie-recommended system in Douban, a well-known social media network in China. This movie network includes objects from six types of entities (e.g., users, movies, groups, actors) and relations between them. Links between objects represent different semantics. For example, links exist between users and users denoting the friendship relations, between users and movies denoting rating and rated relations.

8.2.2.1 Recommendation on Heterogeneous Network

For a target user, recommended systems usually recommend items according to users similar to his/her. In HIN, there are a number of meta paths [13] connecting users, such as "User–User" (UU) and "User–Movie–User" (UMU). Based on these paths, users have different types of similarities. After obtaining the path-based similarity of users, we can recommend items according to the similar users of the target user. More importantly, the meta paths connecting users have different semantics, which can represent different recommendation models. As an example shown in Fig. 8.4, the UMU path means users who view the same movies with the target user. It will recommend movies viewed by users having similar viewing records with the target user. It is collaborative recommendation in essential. Based on the HIN framework, we can flexibly represent different recommendation models through properly setting meta paths. In the following section, we will specifically introduce the semantic recommendation method, where technique details can be found in [12].

8.2.2.2 Semantic Recommendation with Single Path

Based on the path-based similarity of users, we find the similar users of a target user under a given path, and then the rating score of the target user on an item can be inferred according to the rating scores of his similar users on the item. Assume that the range of rating scores is form 1 to N (e.g., 5); P is a set of meta paths; $R \in \mathbf{R}^{|U|\times|I|}$ is the rating matrix, where $R_{u,i}$ denotes the rating score of user u on item i; and $S \in \mathbf{R}^{|U|\times|U|}$ is the path-based similarity matrix of users, where $S^{(l)}_{u,v}$ is the similarity of users u and v under path P_l. Note that the similarity matrix can be calculated offline with some path-based similarity measures [13]. Under a meta path P_l, the predicted rating score of a user u on an item i denoted as $\hat{R}^{(l)}_{u,i}$ is:

$$\hat{R}^{(l)}_{u,i} = \frac{\sum_{v=1}^{|U|} S^{(l)}_{u,v} \times R_{v,i}}{\sum_{v=1}^{|U|} S^{(l)}_{u,v}}. \tag{8.7}$$

According to Eq. 8.7, we can predict the rating score of a user on an item under a given path, and then recommend the item with the high score for a target user.

8.2.2.3 Hybrid Recommendation with Multiple Paths

Under different meta paths, there are different predicted rating scores. In order to calculate the composite score, we employ a personalized weight learning method with weight regularization [12]. As we know, many users have the similar interest preferences, that is, we assume that two similar users have consistent weight preferences on meta paths. For users with little rating information, their path weights can be learnt from the weights of their similar users, since the similarity information of users are more available through meta paths. So we design a weight regularization term, which compels the weights of a user to be consistent to the average of weights of his similar users. The weight matrix is denoted as $W \in R^{|U| \times |P|}$, in which each entry, denoted as $W^{(l)}_u$, means the preference weight of user u on path P_l. The column vector $W^l \in R^{|U| \times 1}$ means the weight vector of all users on path P_l. The following optimization function can learn users' preference weight W.

$$\begin{aligned} \min_W \mathrm{L}(W) = & \ \tfrac{1}{2}||I \odot (R - \sum_{l=1}^{|P|} diag(W^{(l)})\hat{R}^{(l)})||_2^2 \\ + & \ \tfrac{\lambda_1}{2} \sum_{l=1}^{|P|} ||W^{(l)} - \bar{S}^{(l)} W^{(l)}||_2^2 + \tfrac{\lambda_0}{2}||W||_2^2 \\ s.t. & \quad W \geq 0. \end{aligned} \tag{8.8}$$

where $\bar{S}^{(l)}_{u,v} = \frac{S^{(l)}_{u,v}}{\sum_v S^{(l)}_{u,v}}$ is the normalized user similarity based on path P_l, I is an indicator matrix with $I_{u,i} = 1$ if user u rated item i, and otherwise $I_{u,i} = 0$, the notation $\odot$ is the Hadamard product between matrices, and $diag(W^{(l)})$ means the diagonal matrix transformed from a vector $W^{(l)}$.

And thus, the predicted rating $\hat{R}_{u,i}$ of user u rating item i under all paths is as follows:

$$\hat{R}_{u,i} = \sum_{l=1}^{|P|} W^{(l)}_u \times \hat{R}^{(l)}_{u,i}. \tag{8.9}$$

The hybrid recommendation results combine the recommendation from multiple meta paths, and the weight matrix W records the user preferences on these paths. So we can explain the recommendation results according to user path preferences and semantics containing in each path.

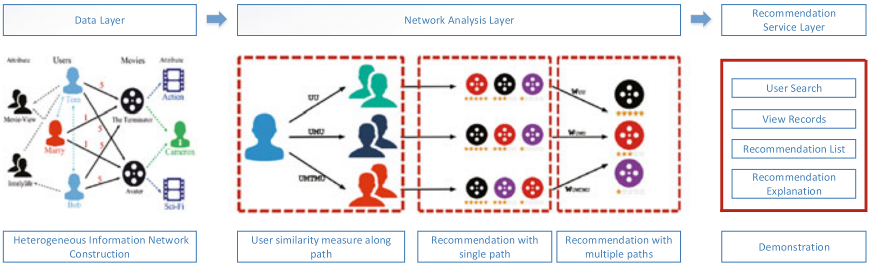


Fig. 8.5 The architecture of *RecExp* system

8.2.3 System Framework

According to the HIN-based recommendation method introduced above, we design the *RecExp* system. Figure 8.5 shows the system architecture. The three main components are detailed as follows:

- Data layer: It extracts data from different data sources (e.g., database and Web) to construct an HIN. Figure 8.4 shows the network schema of HIN in our movie-recommended system demo.
- Network analysis layer: It analyzes the HIN and provides the recommendation services. As we have illustrated in the above section, it first computes the similarities between users along different meta paths, such as “User–Movie–User.” And then, based on similarity of users, we find the similar users of a target user under a given path, and the predicted rating score of the target user on a movie can be inferred from the rating scores of these similar users on the movie. Under different meta paths, there are different predicted rating scores. Through the weight learning method, we assign each meta path with a preference weight for each user, and the final predicted rating under all meta paths can be the weighted average of predicted rating under each meta path.
- Recommendation service layer: It provides the succinct and friendly Web interface of recommendation services. The recommendation services include five kinds of semantic recommendations through setting different meta paths, hybrid recommendation with explanation, and the view record for the searched user.

8.2.4 System Demonstration

Figure 8.6 demonstrates the interface of the *RecExp* system. It consists of five major components:

- **Search box**: Users can input a certain user ID in the search box.

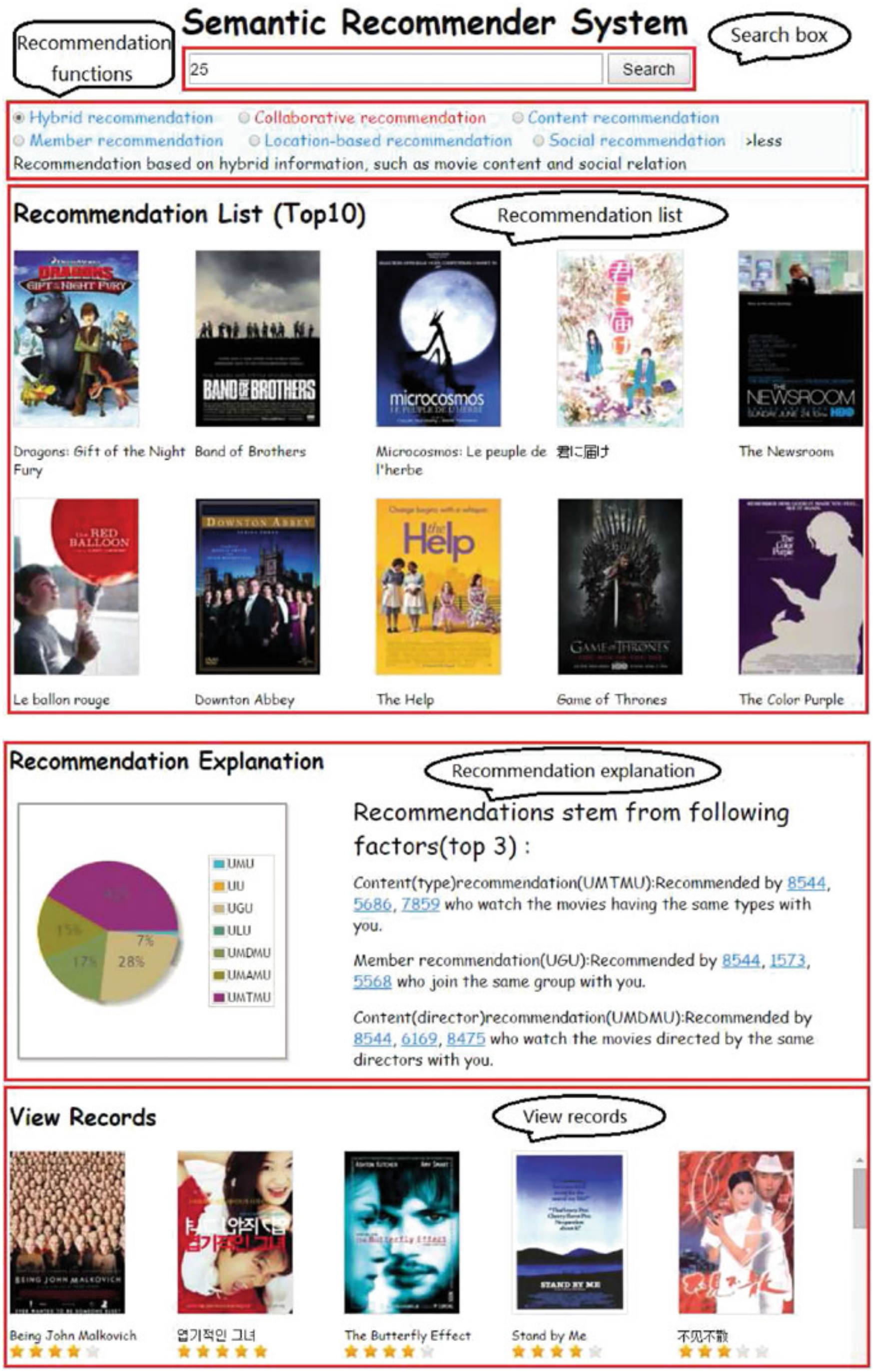


Fig. 8.6 The *RecExp* system

Table 8.1 The recommendation results of different recommendation functions

Recommendation model	Top five recommendation results
Collaborative recommendation	Sherlock, 127 Hours, Game of Thrones, Taxi Driver, The Crucible
Content recommendation	The Big Bang Theory 2, Once a Thief, The Big Bang Theory 4, 2 Broke Girls, The Monkey King
Member recommendation	The cove, Detachment, Inglorious Basterds, The Lives of Others, All About Lily
Location-based recommendation	Farewell My Concubine, Nuovo Cinema Paradise, The Cove, Saving Private Ryan, Sherlock
Social recommendation	Spirited Away, The Pursuit of Happiness, Edward, Scissor Hands

- **Recommendation functions**. There are six recommendation function buttons. Each function button represents a typical recommendation model through selecting a meta path. For example, the collaborative filtering corresponds to the UMU path, and the social recommendation corresponds to the UU path. The description of the selected recommendation model is detailed under the button box. For example, if you press the "Hybrid recommendation" button, the below panel will show "Recommendation based on hybrid information, such as movie content and social relation."
- **Recommendation list**. It shows the top 10 results recommended by the recommendation method you select.
- **Recommendation explanation**: The function will be invoked when the "Hybrid recommendation" function is selected. Since the hybrid recommendation generates the results through multiple meta paths, the fan chart shows the weights of each meta path which can represent the user preference on these paths. The larger the weight is, the more the user prefers to get recommendation from the corresponding meta path. On the right of the fan chart, it shows three most important meta paths and corresponding explanations. In each explanation, we display the three most similar users with the target user based on corresponding meta path.
- **View records**: It displays the view records of a certain user.

We showcase the *RecExp* prototype system using Douban movie data as the example application. The Douban movie data was downloaded from Douban Web site.[2] The dataset includes 13,367 users and 12,677 movies with 1,068,278 movie ratings ranging from 1 to 5, which are organized as a star-schema HIN shown in Fig. 8.4. The dataset includes the social relations among users and the attribute information of users and movies. With this dataset, we will illustrate two major functions: semantic recommendation and hybrid recommendation with explanation.

In the semantic recommendation, users can specify a user ID and the recommendation model such as collaborative recommendation, the recommendation results

[2]www.douban.com/.

will be exhibited in the below panel. For example, we specify user 25 and select five different recommendation functions, whose recommendation results are shown in Table 8.1. We can see that the recommendation results are different based on different meta paths. Different users have their personalized preferences. Through setting proper recommendation model, users can find their own favorite movies. For example, if a user prefers to get new movies by friends' recommendations, he can choose the social recommendation.

When we select the "Hybrid recommendation" function, the system will recommend a composite results stemming from five semantic recommendation models and display top 10 recommendation. Moreover, the recommendation explanation box will explain the recommendation reasons.

For example, we search user 25 and select the "Hybrid recommendation" function, the system will show the recommendation results and give the recommendation explanation shown in Fig. 8.6. In this case, the UMTMU path has the largest weight which means this user has the preference on a certain film type. Among movies that this user has seen, the drama and love movies are his favorite. So these types of movies make the largest proportion in his recommendation list. The system captures this user's preference for film type and displays it in the fan chart. In addition, the system displays three explanations corresponding to the three most important meta paths. For example, the system will list three most similar users with the same file type taste under the UMTMU path, if they are willing to be shown under privacy agreement.

8.3 Other Prototype Systems on Heterogeneous Network

In the section above, we have introduced two prototype systems for recommendation based on HIN. Besides that, many demo systems have also designed prototype applications on HIN. Yu et al. [23] demonstrate a prototype system on query-driven discovery of semantically similar substructures in heterogeneous networks. Danilevsky et al. [3] present the AMETHYST system for exploring and analyzing a topical hierarchy constructed from an HIN. In LikeMiner system, Jin et al. [6] introduce a heterogeneous network model for social media with "likes," and propose "like" mining algorithms to estimate representativeness and influence of objects. Meanwhile, they design SocialSpamGuard [5], a scalable and online social media spam detection system for social network security. Taking DBLP as an example, Tao et al. [18] construct a Research-Insight system to demonstrate the power of database-oriented information network analysis, including ranking, clustering, classification, recommendation, and prediction. Furthermore, they construct a semi-structured news information network NewsNet and develop a NewsNetExplorer system [19] to provide a set of news information network exploration and mining functions.

Some real application systems have also been designed. One of the most famous works is ArnetMiner [3] [16], which offers comprehensive search and mining services for academic community. ArnetMiner not only provides abundant online academic services but also offers ideal test platform for heterogeneous information network analysis. PatentMiner [4] [17] is another application which is a general topic-driven framework for analyzing and mining heterogeneous patent networks.

8.4 Conclusions

With the surge of heterogeneous information network analysis, many prototype systems, even real systems, have been built based on heterogeneous networks. In this chapter, we introduce two prototype systems for recommendations. These prototype systems illustrate the advantages of heterogeneous information on semantics capture and information integration. However, compared to the boom of research on heterogeneous information network, the real applications are relatively insufficient. In the future, we need to solve practical problems in system construction, such as network construction with noise data, large-scale data processing, and scenario design of novel applications.

References

1. Balabanovic, M., Shoham, Y.: Content-based collaborative recommendation. Commun. ACM **40**(3), 66–72 (1997)
2. Breese, J., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: UAI, pp. 43–52 (1998)
3. Danilevsky, M., Wang, C., Tao, F., Nguyen, S., Chen, G., Desai, N., Wang, L., Han, J.: Amethyst: a system for mining and exploring topical hierarchies of heterogeneous data. In: KDD, pp. 1458–1461 (2013)
4. Han, J.: Mining heterogeneous information networks by exploring the power of links. In: DS, pp. 13–30 (2009)
5. Jin, X., Lin, C.X., Luo, J., Han, J.: Socialspamguard: a data mining-based spam detection system for social media networks. Proc. Vldb Endow. **4**(12), 1458–1461 (2011)
6. Jin, X., Wang, C., Luo, J., Yu, X., Han, J.: LikeMiner: a system for mining the power of 'like' in social media networks. In: KDD, pp. 753–756 (2011)
7. Lao, N., Cohen, W.: Fast query execution for retrieval models based on path constrained random walks. In: KDD, pp. 881–888 (2010)
8. Shang, M.S., Lu, L., Zhang, Y.C., Zhou, T.: Empirical analysis of web-based user-object bipartite networks. In: EPL, vol. 90(0120), p. 48006 (2010)
9. Sharma, A., Cosley, D.: Do social explanations work? Studying and modeling the effects of social explanations in recommender systems. In: WWW, pp. 1133–1143 (2013)
10. Shi, C., Kong, X., Yu, P.S., Xie, S., Wu, B.: Relevance search in heterogeneous networks. In: International Conference on Extending Database Technology, pp. 180–191 (2012)

[3] http://aminer.org/.

[4] http://pminer.org/home.do?m=home.

11. Shi, C., Li, Y., Zhang, J., Sun, Y., Yu, P.S.: A survey of heterogeneous information network analysis. Comput. Sci. **134**(12), 87–99 (2015)
12. Shi, C., Zhang, Z., Luo, P., Yu, P.S., Yue, Y., Wu, B.: Semantic path based personalized recommendation on weighted heterogeneous information networks. In: The ACM International, pp. 453–462 (2015)
13. Sun, Y., Han, J.: Mining heterogeneous information networks: a structural analysis approach. SIGKDD Explor. **14**(2), 20–28 (2012)
14. Sun, Y.Z., Han, J.W., Yan, X.F., Yu, P.S., Wu, T.: PathSim: meta path-based top-K similarity search in heterogeneous information networks. In: VLDB, pp. 992–1003 (2011)
15. Symeonidis, P., Nanopoulos, A., Manolopoulos, Y.: Moviexplain: a recommender system with explanations. In: RecSys, pp. 317–320 (2009)
16. Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., Su, Z.: ArnetMiner: extraction and mining of academic social networks. In: KDD, pp. 990–998 (2008)
17. Tang, J., Wang, B., Yang, Y., Hu, P., Zhao, Y., Yan, X., Gao, B., Huang, M., Xu, P., Li, W., Others: PatentMiner: topic-driven patent analysis and mining. In: KDD, pp. 1366–1374 (2012)
18. Tao, F., Yu, X., Lei, K.H., Brova, G., Cheng, X., Han, J., Kanade, R., Sun, Y., Wang, C., Wang, L., Others: Research-insight: providing insight on research by publication network analysis. In: SIGMOD, pp. 1093–1096 (2013)
19. Tao, F., Brova, G., Han, J., Ji, H., Wang, C., Norick, B., El-Kishky, A., Liu, J., Ren, X., Sun, Y.: NewsNetExplorer: automatic construction and exploration of news information networks. In: SIGMOD, pp. 1091–1094 (2014)
20. Tintarev, N., Masthoff, J.: A survey of explanations in recommender systems. In: ICDE Workshop, pp. 801–810 (2007)
21. Vig, J., Sen, S., Riedl, J.: Tagsplanations: explaining recommendations using tags. In: IUI, pp. 47–56 (2009)
22. Wang, B., Ester, M., Bu, J., Cai, D.: Who also likes it? Generating the most persuasive social explanations in recommender systems. In: AAAI, pp. 173–179 (2014)
23. Yu, X., Sun, Y., Zhao, P., Han, J.: Query-driven discovery of semantically similar substructures in heterogeneous networks. In: KDD, pp. 1500–1503 (2012)